**Listning program**

1. Acvitiy\_hitung.java

**package** rahmawati.paerah.jaluralternatif;  
  
**import** android.Manifest;  
**import** android.app.AlertDialog;  
**import** android.app.Dialog;  
**import** android.app.ProgressDialog;  
**import** android.content.Context;  
**import** android.content.DialogInterface;  
**import** android.content.Intent;  
**import** android.content.pm.PackageManager;  
**import** android.graphics.Bitmap;  
**import** android.graphics.BitmapFactory;  
**import** android.graphics.Color;  
**import** android.location.Criteria;  
**import** android.location.Location;  
**import** android.location.LocationListener;  
**import** android.location.LocationManager;  
**import** android.os.AsyncTask;  
**import** android.os.Build;  
**import** android.os.Bundle;  
**import** android.support.annotation.NonNull;  
**import** android.support.annotation.Nullable;  
**import** android.support.v4.app.ActivityCompat;  
**import** android.support.v4.app.FragmentActivity;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.util.Log;  
**import** android.view.View;  
**import** android.webkit.JavascriptInterface;  
**import** android.webkit.WebSettings;  
**import** android.webkit.WebView;  
**import** android.widget.AdapterView;  
**import** android.widget.ArrayAdapter;  
**import** android.widget.Spinner;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** com.akexorcist.googledirection.DirectionCallback;  
**import** com.akexorcist.googledirection.GoogleDirection;  
**import** com.akexorcist.googledirection.constant.TransportMode;  
**import** com.akexorcist.googledirection.constant.Unit;  
**import** com.akexorcist.googledirection.model.Direction;  
**import** com.akexorcist.googledirection.model.Info;  
**import** com.akexorcist.googledirection.model.Leg;  
**import** com.akexorcist.googledirection.model.Route;  
**import** com.akexorcist.googledirection.util.DirectionConverter;  
**import** com.google.android.gms.common.ConnectionResult;  
**import** com.google.android.gms.common.GooglePlayServicesUtil;  
**import** com.google.android.gms.common.api.GoogleApiClient;  
**import** com.google.android.gms.maps.CameraUpdateFactory;  
**import** com.google.android.gms.maps.GoogleMap;  
**import** com.google.android.gms.maps.OnMapReadyCallback;  
**import** com.google.android.gms.maps.SupportMapFragment;  
**import** com.google.android.gms.maps.model.BitmapDescriptor;  
**import** com.google.android.gms.maps.model.BitmapDescriptorFactory;  
**import** com.google.android.gms.maps.model.LatLng;  
**import** com.google.android.gms.maps.model.Marker;  
**import** com.google.android.gms.maps.model.MarkerOptions;  
**import** com.google.android.gms.maps.model.PolylineOptions;  
  
**import** org.json.JSONArray;  
**import** org.json.JSONObject;  
  
**import** java.util.ArrayList;  
**import** java.util.HashMap;  
  
**import** rahmawati.paerah.jaluralternatif.data\_jalur.Data\_jalur;  
  
**public class** Activity\_hitung **extends** AppCompatActivity **implements** GoogleApiClient.OnConnectionFailedListener,  
 LocationListener, GoogleMap.OnMarkerClickListener, OnMapReadyCallback, GoogleApiClient.ConnectionCallbacks, DirectionCallback {  
  
 **final int RQS\_GooglePlayServices** = 1;  
 GoogleMap **gMap**;  
 LatLng **gorsouth** = **new** LatLng(0.534298014805243, 123.0532445758581);  
 **double latitude**, **longitude**;  
 ProgressDialog **pDialog**;  
 ArrayList<HashMap<String, String>> **dataList** = **new** ArrayList<HashMap<String, String>>();  
 JSONArray **college** = **null**;  
 ConnectionDetector **cd**;  
 Boolean **isInternetPresent** = **false**;  
 AlertDialogManager **alert** = **new** AlertDialogManager();  
 HashMap<String, String> **map**;  
 JSONObject **jsonobject**;  
 JSONArray **jsonarray**;  
 ProgressDialog **mProgressDialog**;  
 ArrayList<String> **listnamadealer**;  
 ArrayList<Data\_jalur> **dealersimpan**;  
 **private** String[] **colors** = {**"#7fff7272"**, **"#7f31c7c5"**, **"#7fff8a00"**};  
 LatLng **tujuan**;  
 Marker **markeranda** = **null**;  
 **int check**;  
 CONFIG\_URL **ip** ;  
 WebView **web** ;  
 LatLng **posisi**;  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***cari\_jarak***);  
  
  
 **web** = (WebView)findViewById(R.id.***web***);  
 WebSettings webSettings = **web**.getSettings();  
 webSettings.setJavaScriptEnabled(**true**);  
  
 SupportMapFragment mapFragment =  
 (SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.***mapcari***);  
 mapFragment.getMapAsync(**this**);  
  
 *// Download JSON file AsyncTask  
 //location* cekInternet();  
  
 **try** {  
 CekGPS();  
 }**catch** (Exception e){  
 Log.*e*(**"is"**, **"onCreate: "**+ e);  
 }  
 }  
  
  
 @Override  
 **public void** onLocationChanged(Location location) {  
 handleNewLocation(location);  
 }  
  
 @Override  
 **public void** onStatusChanged(String provider, **int** status, Bundle extras) {  
  
 }  
  
 @Override  
 **public void** onProviderEnabled(String provider) {  
  
 }  
  
 @Override  
 **public void** onProviderDisabled(String provider) {  
  
 }  
  
 @Override  
 **public boolean** onMarkerClick(Marker marker) {  
 **return false**;  
 }  
  
 **private void** handleNewLocation(Location location) {  
 Log.*d*(**"is "**, location.toString());  
  
 **double** currentLatitude = location.getLatitude();  
 **double** currentLongitude = location.getLongitude();  
 **posisi** = **new** LatLng(currentLatitude, currentLongitude);  
 *//gambar modifikasi marker  
 /\*Bitmap img = BitmapFactory.decodeResource(getResources(),R.drawable.ic\_me);  
 BitmapDescriptor bitmapDescriptor = BitmapDescriptorFactory.fromBitmap(img);  
 //gambar default dari googlemaps  
 BitmapDescriptor defaulticon = BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE\_AZURE);  
 MarkerOptions options = new MarkerOptions()  
 .position(posisi)  
 .title("anda disini ")  
 .icon(defaulticon);\*/* **try** {  
 **markeranda**.remove();  
 } **catch** (NullPointerException e) {  
  
 }  
 *// markeranda = gMap.addMarker(options);* **gMap**.moveCamera(CameraUpdateFactory.*newLatLng*(**posisi**));  
 }  
 @Override  
 **public void** onMapReady(GoogleMap googleMap) {  
 **gMap** = googleMap;  
  
 **gMap**.setMapType(GoogleMap.***MAP\_TYPE\_NORMAL***);  
 **gMap**.moveCamera(CameraUpdateFactory.*newLatLng*(**gorsouth**));  
 **if** (ActivityCompat.*checkSelfPermission*(**this**, Manifest.permission.***ACCESS\_FINE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED*** && ActivityCompat.*checkSelfPermission*(**this**, Manifest.permission.***ACCESS\_COARSE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED***) {  
 *//* ***TODO: Consider calling*** *// ActivityCompat#requestPermissions  
 // here to request the missing permissions, and then overriding  
 // public void onRequestPermissionsResult(int requestCode, String[] permissions,  
 // int[] grantResults)  
 // to handle the case where the user grants the permission. See the documentation  
 // for ActivityCompat#requestPermissions for more details.* **if** (ActivityCompat.*shouldShowRequestPermissionRationale*(Activity\_hitung.**this**,  
 Manifest.permission.***ACCESS\_FINE\_LOCATION***)) {  
  
 *// Show an expanation to the user \*asynchronously\* -- don't block  
 // this thread waiting for the user's response! After the user  
 // sees the explanation, try again to request the permission.* } **else** {  
  
 *// No explanation needed, we can request the permission.* **if** (Build.VERSION.***SDK\_INT*** >= Build.VERSION\_CODES.***M***) {  
 requestForSpecificPermission();  
 }  
  
  
 *// MY\_PERMISSIONS\_REQUEST\_READ\_CONTACTS is an  
 // app-defined int constant. The callback method gets the  
 // result of the request.* }  
  
  
 **return**;  
 }  
 **gMap**.setMyLocationEnabled(**true**);  
 **gMap**.setTrafficEnabled(**true**);  
 **gMap**.setIndoorEnabled(**true**);  
 **gMap**.setBuildingsEnabled(**true**);  
 **gMap**.getUiSettings().setZoomControlsEnabled(**true**);  
 }  
  
 **public void** onRequestPermissionsResult(**int** requestCode,  
 String permissions[], **int**[] grantResults) {  
 **switch** (requestCode) {  
 **case** 101: {  
 *// If request is cancelled, the result arrays are empty.* **if** (grantResults.**length** > 0  
 && grantResults[0] == PackageManager.***PERMISSION\_GRANTED***) {  
  
 *// permission was granted, yay! Do the  
 // contacts-related task you need to do.* } **else** {  
 Toast.*makeText*(**this**, **"berikan ijin untuk lokasi GPS pada perijinan"**, Toast.***LENGTH\_SHORT***).show();  
 finish();  
 *// permission denied, boo! Disable the  
 // functionality that depends on this permission.* }  
 **return**;  
 }  
  
 *// other 'case' lines to check for other  
 // permissions this app might request* }  
 }  
 **private void** requestForSpecificPermission() {  
 ActivityCompat.*requestPermissions*(**this**, **new** String[]{Manifest.permission.***ACCESS\_WIFI\_STATE***,  
 Manifest.permission.***ACCESS\_NETWORK\_STATE***, Manifest.permission.***ACCESS\_FINE\_LOCATION***,  
 Manifest.permission.***ACCESS\_COARSE\_LOCATION***, Manifest.permission.***READ\_PHONE\_STATE***,  
 Manifest.permission.***INTERNET***, Manifest.permission.***READ\_EXTERNAL\_STORAGE***,  
 Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***}, 101);  
 }  
 **public void** CekGPS() {  
 **try** {  
 LocationManager manager = (LocationManager) getSystemService(Context.***LOCATION\_SERVICE***);  
 **if** (!manager.isProviderEnabled(LocationManager.***GPS\_PROVIDER***)) {  
 AlertDialog.Builder builder = **new** AlertDialog.Builder(**this**);  
 builder.setTitle(**"info"**);  
 builder.setMessage(**"Apakah anda akan mengaktifkan GPS?"**);  
 builder.setPositiveButton(**"Ya"**,  
 **new** DialogInterface.OnClickListener() {  
  
 @Override  
 **public void** onClick(DialogInterface arg0, **int** arg1) {  
 *//* ***TODO Auto-generated method stub*** Intent i = **new** Intent(  
 android.provider.Settings.***ACTION\_LOCATION\_SOURCE\_SETTINGS***);  
 startActivity(i);  
  
 }  
 });  
 builder.setNegativeButton(**"Tidak"**,  
 **new** DialogInterface.OnClickListener() {  
  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** arg1) {  
 *//* ***TODO Auto-generated method stub*** dialog.dismiss();  
 }  
 });  
 builder.create().show();  
 }  
 } **catch** (Exception e) {  
 *//* ***TODO: handle exception*** }  
 **int** status = GooglePlayServicesUtil  
 .*isGooglePlayServicesAvailable*(getBaseContext());  
 **if** (status != ConnectionResult.***SUCCESS***) {  
 **int** requestCode = 10;  
 Dialog dialog = GooglePlayServicesUtil.*getErrorDialog*(status, **this**,  
 requestCode);  
 dialog.show();  
 } **else** {  
  
 Criteria criteria = **new** Criteria();  
 LocationManager locationmanager = (LocationManager) getSystemService(Context.***LOCATION\_SERVICE***);  
 String provider = locationmanager.getBestProvider(criteria, **true**);  
 **if** (ActivityCompat.*checkSelfPermission*(**this**, Manifest.permission.***ACCESS\_FINE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED*** && ActivityCompat.*checkSelfPermission*(**this**, Manifest.permission.***ACCESS\_COARSE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED***) {  
 *//* ***TODO: Consider calling*** *// ActivityCompat#requestPermissions  
 // here to request the missing permissions, and then overriding  
 // public void onRequestPermissionsResult(int requestCode, String[] permissions,  
 // int[] grantResults)  
 // to handle the case where the user grants the permission. See the documentation  
 // for ActivityCompat#requestPermissions for more details.* **return**;  
 }  
 Location location = locationmanager.getLastKnownLocation(provider);  
  
 **if** (location != **null**) {  
 onLocationChanged(location);  
 }  
  
 locationmanager.requestLocationUpdates(provider, 500, 0, **this**);  
 **posisi** = **new** LatLng(**latitude**, **longitude**);  
 **try**{  
 **gMap**.animateCamera(CameraUpdateFactory.*newLatLngZoom*(**posisi**,  
 12));}  
 **catch** (Exception e){  
 Log.*e*( **"is"**,**"CekGPS: "** + e );  
 }  
 **gMap**.setOnMarkerClickListener(**this**);  
 }  
 }  
  
  
 @Override  
 **public void** onConnected(@Nullable Bundle bundle) {  
  
 }  
  
 @Override  
 **public void** onConnectionSuspended(**int** i) {  
  
 }  
 **public class** JavaInterface {  
  
 String **x1**, **y1**,**x2**,**y2**,**x**,**y** ;  
  
 **public** JavaInterface(String x1, String y1,String x2,String y2) {  
 **this**.**x1** = x1;  
 **this**.**y1** = y1;  
 **this**.**x2** = x2;  
 **this**.**y2** = y2;  
 **x**=x1+**","**+y1;  
 **y**=x2+**","**+y2;  
 }  
  
  
  
 @JavascriptInterface  
 **public void** showToast() {  
 Toast.*makeText*(getApplicationContext(), **x**, Toast.***LENGTH\_SHORT***).show();  
 }  
 @JavascriptInterface  
 **public** String getpos() {  
  
 **return x**;  
 }  
 @JavascriptInterface  
 **public** String gettuju() {  
  
 **return y**;  
  
 }  
 }  
 @Override  
 **public void** onDirectionSuccess(Direction direction, String rawBody) {  
 **if** (direction.isOK()) {  
  
 **try** {  
 *// gMap.clear();  
 // markeranda.remove();* **gMap**.addMarker(**new** MarkerOptions().position(**posisi**).icon(BitmapDescriptorFactory.*fromResource*(R.drawable.***ic\_me***)));  
 **gMap**.addMarker(**new** MarkerOptions().position(**tujuan**).icon(BitmapDescriptorFactory.*fromResource*(R.drawable.***jaluricon***)));  
 *// for (int i = 0; i < direction.getRouteList().size(); i++) {* Route route = direction.getRouteList().get(0);  
  
 ArrayList<LatLng> directionPositionList = direction.getRouteList().get(0).getLegList().get(0).getDirectionPoint();  
 **gMap**.addPolyline(DirectionConverter.*createPolyline*(**this**, directionPositionList, 5, Color.***RED***));  
  
  
 **gMap**.animateCamera(CameraUpdateFactory.*newLatLngZoom*(**tujuan**, 13));  
 Leg leg = route.getLegList().get(0);  
 Info distanceInfo = leg.getDistance();  
 Info durationInfo = leg.getDuration();  
 String distance = distanceInfo.getText();  
 String duration = durationInfo.getText();  
  
 **double** la1 = **posisi**.**latitude**;  
 **double** lo1 = **posisi**.**longitude**;  
 **double** la2 = **tujuan**.**latitude**;  
 **double** lo2 = **tujuan**.**longitude**;  
  
 **web**.addJavascriptInterface(**new** JavaInterface(String.*valueOf*(la1), String.*valueOf*(lo1), String.*valueOf*(la2), String.*valueOf*(lo2)), **"android"**);  
 **web**.loadUrl(**"file:///android\_asset/map.html?posisi="** + **posisi** + **"&tujuan="** + **tujuan**);  
 Toast.*makeText*(**this**, String.*valueOf*(**posisi**), Toast.***LENGTH\_SHORT***).show();  
  
  
 } **catch** (Exception e) {  
 Log.*e*(**"is"**, **"gagal : "** + e);  
 }  
 }  
 }  
  
 @Override  
 **public void** onDirectionFailure(Throwable t) {  
  
 }  
  
 @Override  
 **public void** onConnectionFailed(@NonNull ConnectionResult connectionResult) {  
  
 }  
  
 **private class** DownloadJSON **extends** AsyncTask<Void, Void, Void> {  
  
 @Override  
 **protected** Void doInBackground(Void... params) {  
 *// Locate the WorldPopulation Class* **dealersimpan** = **new** ArrayList<Data\_jalur>();  
 *// Create an array to populate the spinner* **listnamadealer** = **new** ArrayList<String>();  
  
  
 *// JSON file URL address* String url;  
 **ip** = **new** CONFIG\_URL();  
 url = **ip**.getUrljalur();  
  
 JSONParser jParser = **new** JSONParser();  
  
 **jsonobject** = jParser.getJSONFromUrl(url);  
  
 **try** {  
 *// Locate the NodeList name* **jsonarray** = **jsonobject**.getJSONArray(**"jalur"**);  
 **for** (**int** i = 0; i < **jsonarray**.length(); i++) {  
 **jsonobject** = **jsonarray**.getJSONObject(i);  
  
 Data\_jalur datade = **new** Data\_jalur();  
  
 datade.setId(**jsonobject**.optString(**"id"**));  
 datade.setNama(**jsonobject**.optString(**"nama"**));  
 datade.setLat(**jsonobject**.optString(**"lati"**));  
 datade.setLongi(**jsonobject**.optString(**"longi"**));  
 **dealersimpan**.add(datade);  
  
 *// Populate spinner with country names* **listnamadealer**.add(**jsonobject**.optString(**"nama"**));  
  
 }  
 } **catch** (Exception e) {  
 Log.*e*(**"Error"**, e.getMessage());  
 e.printStackTrace();  
 }  
 **return null**;  
 }  
  
 @Override  
 **protected void** onPostExecute(Void args) {  
 *// Locate the spinner in activity\_main.xml* **final** Spinner mySpinner = (Spinner) findViewById(R.id.***spinjalur***);  
  
 *// Spinner adapter* mySpinner.setAdapter(**new** ArrayAdapter<String>(Activity\_hitung.**this**,  
 android.R.layout.***simple\_spinner\_dropdown\_item***,  
 **listnamadealer**));  
 mySpinner.setSelection(0);  
 *// Spinner on item click listener* mySpinner  
 .setOnItemSelectedListener(**new** AdapterView.OnItemSelectedListener() {  
  
 @Override  
 **public void** onItemSelected(AdapterView<?> arg0,  
 View arg1, **int** position, **long** arg3) {  
 *//* ***TODO Auto-generated method stub*** *// Locate the textviews in activity\_main.xml  
 /\* TextView txtrank = (TextView) findViewById(R.id.rank);  
 TextView txtcountry = (TextView) findViewById(R.id.country);  
 TextView txtpopulation = (TextView) findViewById(R.id.population);  
  
 // Set the text followed by the position  
 txtrank.setText("Rank : "  
 + world.get(position).getRank());  
 txtcountry.setText("Country : "  
 + world.get(position).getCountry());  
 txtpopulation.setText("Population : "  
 + world.get(position).getPopulation());\*/* **check** = **check** + 1;;  
 **if**(**check**>1)  
 {  
 **try**{ **gMap**.clear();  
 }**catch** (Exception e){  
 Log.*e*(**"is"**, **"hapus marker gagal "** + e );  
 }  
  
 **tujuan** = **new** LatLng(Double.*parseDouble*(**dealersimpan**.get(position).getLat()), Double.*parseDouble*(**dealersimpan**.get(position).getLongi()));  
  
 **double** lata = **posisi**.**latitude**;  
 **double** latb= **tujuan**.**latitude**;  
 **double** longa = **posisi**.**longitude**;  
 **double** longb= **tujuan**.**longitude**;  
  
  
 GoogleDirection.*withServerKey*(**"AIzaSyCvKWPe2ON1mcIociiwf-InekckM-75rCU"**)  
 .from(**posisi**)  
 .to(**tujuan**)  
 .transportMode(TransportMode.***DRIVING***)  
 .unit(Unit.***METRIC***)  
 .alternativeRoute(**true**)  
 .execute(Activity\_hitung.**this**);  
  
 }}  
 @Override  
 **public void** onNothingSelected(AdapterView<?> arg0) {  
  
 }  
 });  
 }  
 }  
 **public void** cekInternet() {  
 **cd** = **new** ConnectionDetector(getApplicationContext());  
 **isInternetPresent** = **cd**.isConnectingToInternet();  
  
 **if** (**isInternetPresent**) {  
  
 **new** DownloadJSON().execute();  
  
 } **else** {  
  
 **alert**.showAlertDialog(Activity\_hitung.**this**, **"Peringatan"**,  
 **"cek koneksi internet."**, **false**);  
 }  
 }  
}

Menu utama

**package** rahmawati.paerah.jaluralternatif;  
  
**import** android.content.DialogInterface;  
**import** android.content.Intent;  
**import** android.os.Bundle;  
**import** android.support.v7.app.AlertDialog;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.view.LayoutInflater;  
**import** android.view.Menu;  
**import** android.view.MenuItem;  
**import** android.view.View;  
**import** android.widget.Button;  
  
  
  
**public class** menu\_utama **extends** AppCompatActivity {  
  
 Button **a**,**b**,**c**,**d**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_menu\_utama***);  
  
  
  
  
 **a**=(Button)findViewById(R.id.***btinfo***);  
 **b**=(Button)findViewById(R.id.***btdata***);  
 **c**=(Button)findViewById(R.id.***btmap***);  
  
  
 **a**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 Intent i = **new** Intent(menu\_utama.**this**,Tab.**class**);  
 startActivity(i);  
 }  
 });  
 **b**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 Intent i = **new** Intent(menu\_utama.**this**,Activity\_Map.**class**);  
 startActivity(i);  
 }  
 });  
 **c**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 Intent i = **new** Intent(menu\_utama.**this**,Activity\_hitung.**class**);  
 startActivity(i);  
 }  
 });  
  
  
 }  
 **public void** onBackPressed() {  
  
 finish();  
 }  
 @Override  
 **public boolean** onCreateOptionsMenu(Menu menu) {  
 *// Inflate the menu; this adds items to the action bar if it is present.* getMenuInflater().inflate(R.menu.***menu\_tab***, menu);  
 **return true**;  
 }  
  
 @Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
 *// Handle action bar item clicks here. The action bar will  
 // automatically handle clicks on the Home/Up button, so long  
 // as you specify a parent activity in AndroidManifest.xml.* **int** id = item.getItemId();  
  
 *//noinspection SimplifiableIfStatement* **if** (id == R.id.***profil\_set***) {  
  
 AlertDialog.Builder builder = **new** AlertDialog.Builder(**this**);  
 *// Get the layout inflater* LayoutInflater inflater = **this**.getLayoutInflater();  
  
 *// Inflate and set the layout for the dialog  
 // Pass null as the parent view because its going in the dialog layout* builder.setView(inflater.inflate(R.layout.***profil***, **null**))  
 *// Add action buttons* .setNegativeButton(**"Close"**, **new** DialogInterface.OnClickListener() {  
 **public void** onClick(DialogInterface dialog, **int** id) {  
 dialog.dismiss();  
 }  
 })  
 .show();  
 }  
  
 **return super**.onOptionsItemSelected(item);  
 }  
  
}

Activitiy maps

**package** rahmawati.paerah.jaluralternatif;  
  
**import** android.Manifest;  
**import** android.app.AlertDialog;  
**import** android.app.Dialog;  
**import** android.app.ProgressDialog;  
**import** android.content.Context;  
**import** android.content.DialogInterface;  
**import** android.content.Intent;  
**import** android.content.pm.PackageManager;  
**import** android.graphics.Color;  
**import** android.graphics.Typeface;  
**import** android.location.Criteria;  
**import** android.location.Location;  
**import** android.location.LocationListener;  
**import** android.location.LocationManager;  
**import** android.os.AsyncTask;  
**import** android.os.Bundle;  
**import** android.support.v4.app.ActivityCompat;  
**import** android.support.v4.app.FragmentActivity;  
**import** android.util.Log;  
**import** android.view.Gravity;  
**import** android.view.View;  
**import** android.widget.LinearLayout;  
**import** android.widget.TextView;  
  
**import** com.google.android.gms.common.ConnectionResult;  
**import** com.google.android.gms.common.GooglePlayServicesUtil;  
**import** com.google.android.gms.maps.CameraUpdateFactory;  
**import** com.google.android.gms.maps.GoogleMap;  
**import** com.google.android.gms.maps.GoogleMap.OnMarkerClickListener;  
**import** com.google.android.gms.maps.OnMapReadyCallback;  
**import** com.google.android.gms.maps.SupportMapFragment;  
**import** com.google.android.gms.maps.model.BitmapDescriptorFactory;  
**import** com.google.android.gms.maps.model.LatLng;  
**import** com.google.android.gms.maps.model.Marker;  
**import** com.google.android.gms.maps.model.MarkerOptions;  
  
**import** org.json.JSONArray;  
**import** org.json.JSONException;  
**import** org.json.JSONObject;  
  
**import** java.util.ArrayList;  
**import** java.util.HashMap;  
  
**public class** Activity\_Map **extends** FragmentActivity **implements** LocationListener, OnMarkerClickListener, OnMapReadyCallback {  
  
 **final int RQS\_GooglePlayServices** = 1;  
 GoogleMap **gMap**;  
 LatLng **gorsouth** =**new** LatLng(0.5473091, 123.0361635);  
 **double latitude**, **longitude**;  
 ProgressDialog **pDialog**;  
  
 ArrayList<HashMap<String, String>> **dataList** = **new** ArrayList<HashMap<String, String>>();  
  
  
 JSONArray **college** = **null**;  
  
  
 ConnectionDetector **cd**;  
 Boolean **isInternetPresent** = **false**;  
  
 AlertDialogManager **alert** = **new** AlertDialogManager();  
  
 HashMap<String, String> **map**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 *//* ***TODO Auto-generated method stub* super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_mapjalur***);  
  
 cekInternet();  
  
 SupportMapFragment mapFragment =  
 (SupportMapFragment) getSupportFragmentManager().findFragmentById(R.id.***map***);  
 mapFragment.getMapAsync(**this**);  
  
  
  
**try** {  
 CekGPS();  
}**catch** (Exception e){  
  
}  
  
  
  
  
 *// refresh = (Button) findViewById(R.id.reload);  
 // refresh.setOnClickListener(new View.OnClickListener() {  
  
 //@Override  
 //public void onClick(View arg0){  
 // //* ***TODO Auto-generated method stub*** */// cekInternet();  
  
  
 // cari = (Button) findViewById(R.id.cari);  
 // cari.setOnClickListener(new View.OnClickListener() {  
  
 // @Override  
 // public void onClick(View arg0) {  
 // //* ***TODO Auto-generated method stub*** *// Intent i = new Intent(getApplicationContext(), Activity\_TampilanMap.class);  
 // startActivity(i);  
 // }* }  
  
 */\*@Override  
 public boolean onCreateOptionsMenu(Menu menu) {  
 MenuInflater inflater = getMenuInflater();  
 inflater.inflate(R.menu.cari, menu);  
 return true;  
  
 }  
  
 @Override  
 public boolean onOptionsItemSelected(MenuItem item) {  
 switch (item.getItemId()) {  
 case R.id.action\_cari:  
 Intent imap= new Intent(this, Activity\_hitung.class);  
 startActivity(imap);  
 return true;  
  
 default:  
 return super.onOptionsItemSelected(item);  
 }  
  
 }\*/* **public void** onBackPressed() {  
 Intent imap= **new** Intent(**this**, menu\_utama.**class**);  
 startActivity(imap);  
 finish();  
 }  
  
 @Override  
 **public void** onMapReady(GoogleMap googleMap) {  
 **gMap** = googleMap;  
 **gMap**.setMapType(GoogleMap.***MAP\_TYPE\_NORMAL***);  
 **gMap**.moveCamera(CameraUpdateFactory.*newLatLng*(**gorsouth**));  
 **if** (ActivityCompat.*checkSelfPermission*(**this**, Manifest.permission.***ACCESS\_FINE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED*** && ActivityCompat.*checkSelfPermission*(**this**, Manifest.permission.***ACCESS\_COARSE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED***) {  
 *//* ***TODO: Consider calling*** *// ActivityCompat#requestPermissions  
 // here to request the missing permissions, and then overriding  
 // public void onRequestPermissionsResult(int requestCode, String[] permissions,  
 // int[] grantResults)  
 // to handle the case where the user grants the permission. See the documentation  
 // for ActivityCompat#requestPermissions for more details.* **return**;  
 }  
 **gMap**.setInfoWindowAdapter(**new** GoogleMap.InfoWindowAdapter() {  
 @Override  
 **public** View getInfoWindow(Marker marker) {  
  
  
 **return null**;  
 }  
  
 @Override  
 **public** View getInfoContents(Marker marker) {  
 LinearLayout info = **new** LinearLayout(getApplicationContext());  
 info.setOrientation(LinearLayout.***VERTICAL***);  
  
 TextView title = **new** TextView(getApplicationContext());  
 title.setTextColor(Color.***BLACK***);  
 title.setGravity(Gravity.***CENTER***);  
 title.setTypeface(**null**, Typeface.***BOLD***);  
 title.setText(marker.getTitle());  
  
 TextView snippet = **new** TextView(getApplicationContext());  
 snippet.setTextColor(Color.***GRAY***);  
 snippet.setText(marker.getSnippet());  
  
 info.addView(title);  
 info.addView(snippet);  
  
 **return** info;  
  
 }  
 });  
 **gMap**.setMyLocationEnabled(**true**);  
 **gMap**.setTrafficEnabled(**true**);  
 **gMap**.setIndoorEnabled(**true**);  
 **gMap**.setBuildingsEnabled(**true**);  
 **gMap**.getUiSettings().setZoomControlsEnabled(**true**);  
 }  
  
 **public class** AmbilData **extends** AsyncTask<String, String, String> {  
  
 @Override  
 **protected void** onPreExecute() {  
 **super**.onPreExecute();  
 **pDialog** = **new** ProgressDialog(Activity\_Map.**this**);  
 **pDialog**.setMessage(**"Tunggu Sebentar ..."**);  
 **pDialog**.setIndeterminate(**false**);  
 **pDialog**.setCancelable(**true**);  
 **pDialog**.show();  
 }  
  
 @Override  
 **protected** String doInBackground(String... arg0) {  
 String url;  
CONFIG\_URL cu = **new** CONFIG\_URL();  
 url = cu.getUrljalur();  
  
 JSONParser jParser = **new** JSONParser();  
  
 JSONObject json = jParser.getJSONFromUrl(url);  
 **try** {  
 **college** = json.getJSONArray(**"jalur"**);  
 Log.*e*(**"error"**, json.getString(**"success"**));  
  
 **for** (**int** i = 0; i <= **college**.length(); i++) {  
  
 JSONObject c = **college**.getJSONObject(i);  
  
 **map** = **new** HashMap<String, String>();  
  
 String id\_1 = c.getString(**"id"**).trim();  
 String latitude\_1 = c.getString(**"lati"**).trim();  
 String longitude\_1 = c.getString(**"longi"**).trim();  
 String nama\_1 = c.getString(**"nama"**).trim();  
 String jammacet = c.getString(**"jammacet"**).trim();  
  
  
  
 **map**.put(**"id"**, id\_1);  
 **map**.put(**"nama"**, nama\_1);  
 **map**.put(**"jam"**, jammacet);  
 **map**.put(**"lati"**, latitude\_1);  
 **map**.put(**"longi"**, longitude\_1);  
  
  
  
  
  
  
 **dataList**.add(**map**);  
  
 }  
  
 } **catch** (JSONException e) {  
  
 }  
  
 **return null**;  
 }  
  
 @Override  
 **protected void** onPostExecute(String result) {  
 *//* ***TODO Auto-generated method stub* super**.onPostExecute(result);  
 **pDialog**.dismiss();  
 **try** {  
 **for** (**int** x = 0; x < **dataList**.size(); x = x + 1) {  
  
 **double** latasal = Double.*parseDouble*(**dataList**.get(x).get(  
 **"lati"**));  
 **double** longasal = Double.*parseDouble*(**dataList**.get(x).get(  
 **"longi"**));  
 LatLng posisi = **new** LatLng(latasal, longasal);  
 String nama = **dataList**.get(x).get(**"nama"**);  
 String al = **dataList**.get(x).get(**"alamat"**);  
 String jamm = **dataList**.get(x).get(**"jam"**);  
  
  
 Log.*e*(**"is"**, **"latlang "** + posisi );  
  
 **gMap**.addMarker(**new** MarkerOptions()  
 .position(posisi)  
 .title(nama)  
 .snippet(jamm )  
 .icon(BitmapDescriptorFactory.*fromResource*(R.drawable.***jaluricon***)));  
 *// .icon(BitmapDescriptorFactory.defaultMarker(BitmapDescriptorFactory.HUE\_AZURE)));* }  
 }**catch** (Exception e) {  
  
 Log.*e*(**"is"**, **"gagal addmarker "** + e );  
}  
  
  
 }  
  
 }  
  
  
  
  
 @Override  
 **public void** onLocationChanged(Location location) {  
 *//* ***TODO Auto-generated method stub* try** {  
 **latitude** = location.getLatitude();  
 **longitude** = location.getLongitude();  
 } **catch** (Exception e) {  
 *//* ***TODO: handle exception*** }  
 }  
  
 **public void** CekGPS() {  
 **try** {  
 LocationManager manager = (LocationManager) getSystemService(Context.***LOCATION\_SERVICE***);  
 **if** (!manager.isProviderEnabled(LocationManager.***GPS\_PROVIDER***)) {  
 AlertDialog.Builder builder = **new** AlertDialog.Builder(**this**);  
 builder.setTitle(**"info"**);  
 builder.setMessage(**"Apakah anda akan mengaktifkan GPS?"**);  
 builder.setPositiveButton(**"Ya"**,  
 **new** DialogInterface.OnClickListener() {  
  
 @Override  
 **public void** onClick(DialogInterface arg0, **int** arg1) {  
 *//* ***TODO Auto-generated method stub*** Intent i = **new** Intent(  
 android.provider.Settings.***ACTION\_LOCATION\_SOURCE\_SETTINGS***);  
 startActivity(i);  
  
 }  
 });  
 builder.setNegativeButton(**"Tidak"**,  
 **new** DialogInterface.OnClickListener() {  
  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** arg1) {  
 *//* ***TODO Auto-generated method stub*** dialog.dismiss();  
 }  
 });  
 builder.create().show();  
 }  
 } **catch** (Exception e) {  
 *//* ***TODO: handle exception*** }  
 **int** status = GooglePlayServicesUtil  
 .*isGooglePlayServicesAvailable*(getBaseContext());  
 **if** (status != ConnectionResult.***SUCCESS***) {  
 **int** requestCode = 10;  
 Dialog dialog = GooglePlayServicesUtil.*getErrorDialog*(status, **this**,  
 requestCode);  
 dialog.show();  
 } **else** {  
  
 Criteria criteria = **new** Criteria();  
 LocationManager locationmanager = (LocationManager) getSystemService(Context.***LOCATION\_SERVICE***);  
 String provider = locationmanager.getBestProvider(criteria, **true**);  
 **if** (ActivityCompat.*checkSelfPermission*(**this**, Manifest.permission.***ACCESS\_FINE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED*** && ActivityCompat.*checkSelfPermission*(**this**, Manifest.permission.***ACCESS\_COARSE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED***) {  
 *//* ***TODO: Consider calling*** *// ActivityCompat#requestPermissions  
 // here to request the missing permissions, and then overriding  
 // public void onRequestPermissionsResult(int requestCode, String[] permissions,  
 // int[] grantResults)  
 // to handle the case where the user grants the permission. See the documentation  
 // for ActivityCompat#requestPermissions for more details.* **return**;  
 }  
 Location location = locationmanager.getLastKnownLocation(provider);  
   
 **if** (location != **null**) {  
 onLocationChanged(location);  
 }  
   
 locationmanager.requestLocationUpdates(provider, 500, 0, **this**);  
 LatLng posisi = **new** LatLng(**latitude**, **longitude**);  
 **try**{  
 **gMap**.animateCamera(CameraUpdateFactory.*newLatLngZoom*(posisi,  
 12));}  
 **catch** (Exception e){  
 Log.*e*( **"haris"**,**"CekGPS: "** + e );  
 }  
 **gMap**.setOnMarkerClickListener(**this**);  
 }  
 }  
  
 @Override  
 **public void** onProviderDisabled(String provider) {  
 *//* ***TODO Auto-generated method stub*** }  
  
 @Override  
 **public void** onProviderEnabled(String provider) {  
 *//* ***TODO Auto-generated method stub*** }  
  
 @Override  
 **public void** onStatusChanged(String provider, **int** status, Bundle extras) {  
 *//* ***TODO Auto-generated method stub*** }  
  
 **public void** cekInternet() {  
 **cd** = **new** ConnectionDetector(getApplicationContext());  
 **isInternetPresent** = **cd**.isConnectingToInternet();  
  
 **if** (**isInternetPresent**) {  
  
 **new** AmbilData().execute();  
  
 } **else** {  
  
 **alert**.showAlertDialog(Activity\_Map.**this**, **"Peringatan"**,  
 **"cek koneksi internet."**, **false**);  
 }  
 }  
  
 @Override  
 **public boolean** onMarkerClick(Marker marker) {  
 *//* ***TODO Auto-generated method stub*** */\*String id= marker.getId();  
 id = id.substring(1);  
   
 Intent x = new Intent(getApplicationContext(), Detail.class);  
 x.putExtra("nama", dataList.get(Integer.parseInt(id)).get("nama"));  
 x.putExtra("gambar", dataList.get(Integer.parseInt(id)).get("gambar"));  
 x.putExtra("alamat", dataList.get(Integer.parseInt(id)).get("alamat"));  
   
 startActivity(x);\*/* **return false**;  
 }  
  
}